CLAIMS:

1. A computer system with a processing unit and a memory, the processing unit being arranged to fetch memory lines from the memory and execute instructions from the memory lines, each memory line being fetched as a whole and being capable of holding more than one instruction, at least one instruction comprising information that signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line, the processing unit being arranged to respond to the information by controlling said part as signaled by the information.

2. A computer system according to claim 1, wherein the information signals explicitly whether or not the subsequent memory line has to be prefetched during processing of the instruction, the processing unit being arranged to start prefetching of the subsequent memory line in response to the information.

3. A computer system according to Claim 2, wherein the information contains a prefetch bit whose value signals explicitly whether or not the subsequent memory line has to be prefetched.

4. A computer system according to claim 1, wherein the information signals explicitly whether or not an instruction pointer should be updated form a position behind the instruction in the current memory line to a start of the subsequent memory line, so that information following the instruction on the current memory line is skipped over, the processing unit being arranged to update the instruction pointer to the start of the subsequent memory line in response to the information

5. A computer system according to claim 1, wherein the information signals explicitly whether or not processing of the instruction should be stalled, when the instruction is reached from a branching instruction, processing being stalled to fetch the subsequent

memory line that contains a part of the instruction, the processing unit being arranged to stall in response to the information when the instruction is reached from the branching instruction.

6. A computer system according to Claim 1, the processing unit being a VLIW processing unit containing two or more issue slots for issuing operations from the instruction in parallel to the functional units, the instructions being VLIW instructions, capable of containing two or more operations, the instruction comprising a field distinct from the operations to specify said information.

7. A computer system according to Claim 6, the field comprising, in addition to said information, a decompression code that specifies for which issue slots the instruction contains operations.

8. A method of processing instructions in a computer system with a processing unit and a memory, the processing unit being arranged to fetch memory lines from the memory and execute instructions from the memory lines capable of holding more than one instruction, at least one instruction comprising information that signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line, the method comprising

- fetching each memory line as a whole,
- processing an instruction from a current memory line,
- reading the information from the instruction during processing and
- controlling said part as signaled by the information.

9. A method according to claim 8, wherein said controlling comprises at least one of causing a subsequent memory line to be prefetched or a program counter to skip to a start of the subsequent memory line or processing to be stalled when the instruction is reached as a branch target.

10. A computer program comprising instructions for a processor according to claim 1, the instructions comprising said information according to a way the instructions are to be spread over the memory lines.

11. A computer program according to Claim 10, the information in an instruction in a specific memory line comprising at least one of

- an explicit signal to prefetch a subsequent memory line, where an further instruction that follows the instruction is at least partly contained in the subsequent memory line, or
- an explicit signal to realign a program counter to a start of a next memory line when a part of the specific memory line following the instruction contains padding information, or
- an explicit signal to stall processing when the instruction is reached as a branch target, when the instruction extends to the subsequent memory line and is a target of a branch instruction in the computer program.

12. A computer program according to Claim 11, comprising both a first and a second branch target instruction,

- the first branch target instruction being stored in a first memory line and being aligned to a start of the first memory line, the explicit signal to realign being set in a preceding instruction in a preceding memory line that precedes the first memory line, the explicit signal to stall being cleared in the first branch target instruction;
- the second branch target instruction being spread over a second and third memory line, the second branch target instruction immediately following a preceding instruction in the second memory line, the explicit signal to stall being set in the second branch target instruction.

13. A method of generating a program of instructions, at least one instruction comprising a information that signals explicitly how to control a part of processing when a boundary between a current memory line and a subsequent memory line is crossed, the method comprising at least one of:

- inserting, in the information in an instruction that is to be stored in the current memory line, an explicit signal to prefetch the subsequent memory line when a further instruction that follows the instruction is to be stored at least partly in the subsequent memory line; or
- inserting, in the information in an instruction that is to be stored in the current memory line, an explicit signal to realign a program counter to a start of the subsequent memory line when the instruction is followed by padding information in the current memory line; or
- inserting in the information in a branch target instruction that is to be stored spread over the current memory line and the subsequent memory line, an explicit signal to stall processing of the branch target instruction when the instruction is reached as a branch target.

14. A method according to Claim 13, comprising selecting for a branch target instruction, dependent on a frequency of execution of the instruction, whether to
- store the branch target instruction at a start of a memory line, a space in a preceding memory line after a preceding instruction being padded, the explicit signal to realign being set in the preceding instruction;
- store the branch target instruction spread over the memory line and the preceding memory line, the branch target instruction immediately following the preceding instruction, the explicit signal to stall being set in the branch target instruction.

15. A computer program arranged to cause execution of the method according to claim 13.